Package: mFDP (via r-universe)

November 3, 2024

Type Package

Title Flexible Control of the mFDP

Version 0.1.0

Date 2022-08-19

Maintainer Jesse Hemerik <jesse.hemerik@wur.nl>

Description Computes bounds for the median of the false discovery proportion (mFDP). These 50 percent confidence bounds for the FDP are simultaneously valid. The method takes a vector of p-values as input. Also provides mFDP-adjusted p-values. Can be used for flexible mFDP control.

Imports methods

NeedsCompilation no

Author Jesse Hemerik [aut, cre]

Repository CRAN

Date/Publication 2022-08-22 14:30:05 UTC

Contents

`get.bound` *Compute a 50 percent confidence upper bound for the number of false positives*

Description

For a p-value rejection threshold `t`, compute the 50 percent confidence upper bound for the number of false positives. The bounds are simultaneous over `t`.

Usage

```
get.bound(t,c,kappa.max)
```

Arguments

`t`	The p-value threshold
`c`	The tuning parameter, which influences the intercept and slope of the enveloppe. Should be numeric.
`kappa.max`	This value needs to be computes based on the p-values. Together with `c` it defines the bounds.

Value

A non-negative integer, which is a median unbiased (or upward biased) estimate of the number false positives.

Examples

```
#Suppose the envelope that has been computed is defined by c=0.002 and kappa.max=0.001.
#We can then evaluate the envelope at several thresholds t as below.
#This is equivalent to simply entering the formula floor((t+c)/kappa.max).

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.01 is used:
get.bound(t=0.01,c=0.002,kappa.max=0.001)   #12

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.02 is used:
get.bound(t=0.02,c=0.002,kappa.max=0.001)   #22

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.03 is used:
get.bound(t=0.03,c=0.002,kappa.max=0.001)   #32
```

get.kappa.max — *Based on a vector of raw p-values, compute* `kappa.max`*, which defines the mFDP envelope*

Description

Based on a vector of unadjusted(!) p-values, compute `kappa.max`, which together with `c` defines the mFDP envelope

Usage

```
get.kappa.max(P, c="1/(2m)", s1=0, s2=0.1)
```

Arguments

`P`	A vector of p-values.
`c`	The tuning parameter, which influences the intercept and slope of the enveloppe. Should either be numeric or `"1/(2m)"` or `"1/m"`.
`s1`	The smallest p-value threshold of interest. Non-negative.
`s2`	The largest p-value threshold of interest. Should be larger than `s1` and at most 1.

Value

`kappa.max`, which together with `c` defines the mFDP envelope.

Examples

```
set.seed(5193)

### Make some p-values
m=500        #the nr of hypotheses
nrfalse=100  #the nr of false hypotheses

tstats = rnorm(n=m)   #m test statistics
tstats[1:nrfalse] = tstats[1:nrfalse] + 3    #add some signal
P = 1 - pnorm(tstats)                        #compute p-values

### Compute kappa.max. (Taking c to be the default value 1/(2m).)
kappa.max = get.kappa.max(P=P)
kappa.max
```

`mFDP.adjust` *compute mFDP-adjusted p-values*

Description

Provides mFDP-adjusted p-values, given a vector of p-values.

Usage

```
mFDP.adjust(P, c="1/(2m)", s1=0, s2=0.1)
```

Arguments

`P`	A vector of (raw, i.e. unadjusted) p-values.
`c`	The tuning parameter, which influences the intercept and slope of the envelope. Should either be numeric or `"1/(2m)"` or `"1/m"`.
`s1`	The smallest p-value threshold of interest.
`s2`	The largest p-value threshold of interest.

Value

A vector of mFDP-adjusted p-values. Some can be infinity - which can be interpreted as 1.

Examples

```
set.seed(5193)

### make some p-values
m=500        #the nr of hypotheses
nrfalse=100  #the nr of false hypotheses

tstats = rnorm(n=m)   #m test statistics
tstats[1:nrfalse] = tstats[1:nrfalse] + 3    #add some signal
P = 1 - pnorm(tstats)                        #compute p-values

P.adjusted = mFDP.adjust(P=P)    #mFDP-adjusted p-values. Be careful with interpretation.

min(P.adjusted) #0.0208
```

Index